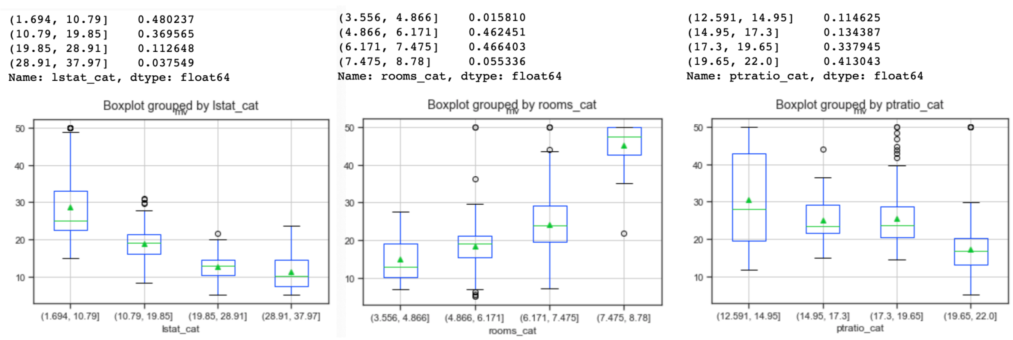
Summary: This paper continues to evaluate 11 features effect on district median property values across 506 samples employing 7 regression methods and various machine learning tools.

Research design: Exploration broadens to all bivariate relationships with the response in order to guide preparation and to set expectations for the results of multiple linear, Ridge, Lasso, ElasticNet, Decision Tree, Random Forest and Gradient Boosting regression models that are fit with varying parameters and ranked by root mean squared errors in cross validation. Test data evaluate the recommended model and a management discussion concludes the report.



Data Exploration / Preparation: (1) **Figures 1a, b, c** above provide box plots of the response on the y-axis binned by highly correlated lstat (-74%), rooms (71%), and ptratio (-49%) on the x-axis. The 1st and 2nd lstat, 3rd and 4th rooms, and 4th ptratio bin separate from the others and each carry significant observation frequencies that may prove these features important in regression results and limit model generalization if train / test splits are not representative.

Table 2b: Differing proportions for random and stratified sampling: ptratio

	Overall	Stratified	Random	Rand. %error	Strat. %error
(12.591, 14.95]	0.114625	0.117647	0.058824	-48.681542	2.636917
(14.95, 17.3]	0.134387	0.137255	0.078431	-41.637832	2.133795
(17.3, 19.65]	0.337945	0.333333	0.401961	18.942782	-1.364522
(19.65, 22.0]	0.413043	0.411765	0.460784	11.558308	-0.309598

Table 2b: Differing proportions for random and stratified sampling: ptratio

	Overall	Stratified	Random	Rand. %error	Strat. %error
(12.591, 14.95]	0.114625	0.117647	0.058824	-48.681542	2.636917
(14.95, 17.3]	0.134387	0.137255	0.078431	-41.637832	2.133795
(17.3, 19.65]	0.337945	0.333333	0.401961	18.942782	-1.364522
(19.65, 22.0]	0.413043	0.411765	0.460784	11.558308	-0.309598

Table 2c: Differing proportions for random and stratified sampling: lstat

	Overall	Stratified	Random	Rand. %error	Strat. %error
(1.694, 10.79]	0.480237	0.480392	0.421569	-12.216574	0.032276
(10.79, 19.85]	0.369565	0.372549	0.421569	14.071511	0.807382
(19.85, 28.91]	0.112648	0.107843	0.117647	4.437564	-4.265566
(28.91, 37.97]	0.037549	0.039216	0.039216	4.437564	4.437564

Tables 2a, b, c, show comparisons of random vs stratified samples indicating that lstat and ptratio might suffer unrepresentative train/test splits, cross validation and final test results.

Table 4:
mv and ptratio modes and counts

	mv mode	count	ptratio mode	count
0	50.0	13.0	20.2	108.0
1	22.0	6.0	14.7	31.0
2	23.1	6.0	21.0	22.0
3	21.7	6.0	17.8	21.0
4	19.3	5.0	17.4	16.0
5	22.2	5.0	19.1	15.0
6	15.6	5.0	18.4	13.0
7	25.0	5.0	18.6	13.0
8	20.6	5.0	16.6	13.0
9	22.9	4.0	19.2	12.0
10	20.1	4.0	15.2	11.0
11	13.4	4.0	13.0	11.0

Figure 3a: mv and rooms

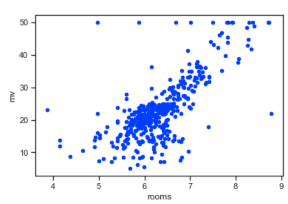


Figure 3b: repeated values in mv and in ptratio

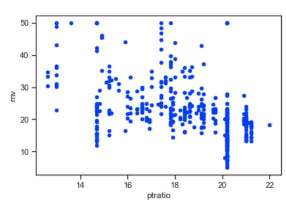
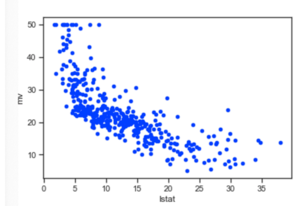


Figure 3c: mv and lstat



Figures 3a,b,c show scatter for these 3 features vs the response, representing the **repeated ptratio** values and **censored mv** values in **table 4** and showing **strong negative lstat-mv**, **strong positive rooms-mv**, and **moderate ptratio-mv** relationships, also numerically found in **table 5**.

Python code: (1) Random sampling splits the dataset 80% train / 20% test. Stratified samples are kept for later testing. (2) A nominal sklearn pipeline is initialized with 7 regression models is loaded with normalizing scaling (for MLR, Ridge, Lasso, and ElasticNet), and a parameter grid. Ridge employs a 'cholesky' solver and alphas = [0,10,100]; Lasso's alphas = [0.1, 1, 10], ElasticNet splits 50/50 between Ridge L2 and Lasso L1 and keeps alphas the same as Lasso. GradientBoostingRegressor=GBR varies learning rate [.1, .5, 1], and together with DecisionTreeRegressor=DT and RandomForestRegressor=RF explore max_depth [1, 2, 3] and max_features [1, log2, 12]. Each is provided 100 estimators per tree which can interact with learning rate to affect GBR overfit. Max_features = 12 provides no randomness in the forest as all features are available and trees will easily fit the data using the most distinctive features. Recommended is log2 and 1 selects just 1 feature which limits variance but increases bias. The results of stratified splits were applied to cross validation in hopes of eeking out higher scores but somehow the code failed (despite hours of work).

The root of -1*neg_mean_squared_error returned by sklearn ranked these 80 model combinations. GradientBoosting filling out the top 10 as shown in **table 6 below**:

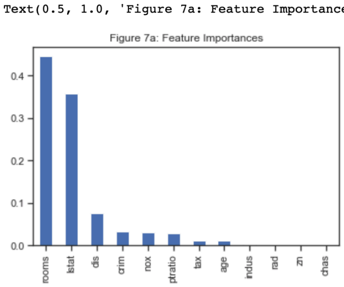
```
Table 6: Model selection
```

rank	model	max_depth	max_feat	learn_rate	train_rmse	test_rmse	train_mse_std	test_mse_std	test_train_ratio
1	([DecisionTreeRegressor(criterion='friedman_ms...	3	12	0.1	1.128008	3.331259	0.099990	3.469513	2.953223
2	([DecisionTreeRegressor(criterion='friedman_ms...	3	log2	0.1	1.366602	3.388887	0.098955	3.834597	2.479791
3	([DecisionTreeRegressor(criterion='friedman_ms...	3	6	0.1	1.202124	3.489852	0.098327	4.189025	2.903073
4	([DecisionTreeRegressor(criterion='friedman_ms...	2	12	0.1	1.904215	3.604883	0.134064	3.797334	1.893107
5	([DecisionTreeRegressor(criterion='friedman_ms...	3	1	0.5	0.618362	3.638868	0.034274	2.807451	5.884688
6	([DecisionTreeRegressor(criterion='friedman_ms...	2	log2	0.1	2.020671	3.653441	0.263365	4.922767	1.808033
7	([DecisionTreeRegressor(criterion='friedman_ms...	2	6	0.1	1.981898	3.661044	0.187054	3.972170	1.847241
8	([DecisionTreeRegressor(criterion='friedman_ms...	2	1	0.5	1.390164	3.697923	0.223190	4.171664	2.660062
9	([DecisionTreeRegressor(criterion='friedman_ms...	3	12	0.5	0.109141	3.739983	0.003090	3.282147	34.267519
10	([DecisionTreeRegressor(criterion='friedman_ms...	3	1	0.1	1.705422	3.763326	0.242889	5.406554	2.206683

Selecting amongst models balances overall scores with a ratio of test-to-train scores measuring each model's generalization. All of the test RMSE = mid 3s which could result from cross validation train/test splitting representativeness. The 5th and 9th models clearly overfit but due to opposite causes related to max_features, as mentioned earlier. Either the 1st or 2nd model is recommended for further testing based on lower rmse and reasonably good generalization. Employing the best model, the 1st, and requesting sklearn to refit to the full train dataset yields feature importance rankings in **figure 7a,b,c** that line up with our expectations for rooms and lstat, and confirm that stratifying train / test by room and lstat is fruitless (2 RHS plots).

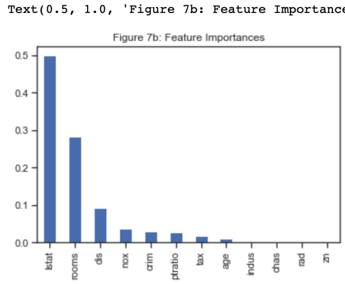
```
GradientBoosting
with room stratified
rmse train = 1.403278420688522
rmse test =  1.5440525180386815

Text(0.5, 1.0, 'Figure 7a: Feature Importances')
```



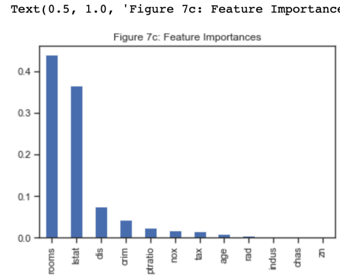

```
GradientBoosting
with room stratified
rmse train = 1.34953519602522
rmse test =  2.50053071738415

Text(0.5, 1.0, 'Figure 7b: Feature Importances')
```

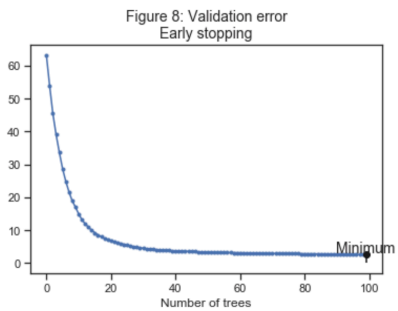



```
GradientBoosting
with lstat stratified
rmse train = 1.34953519602522
rmse test =  3.3747896904981594

Text(0.5, 1.0, 'Figure 7c: Feature Importances')
```

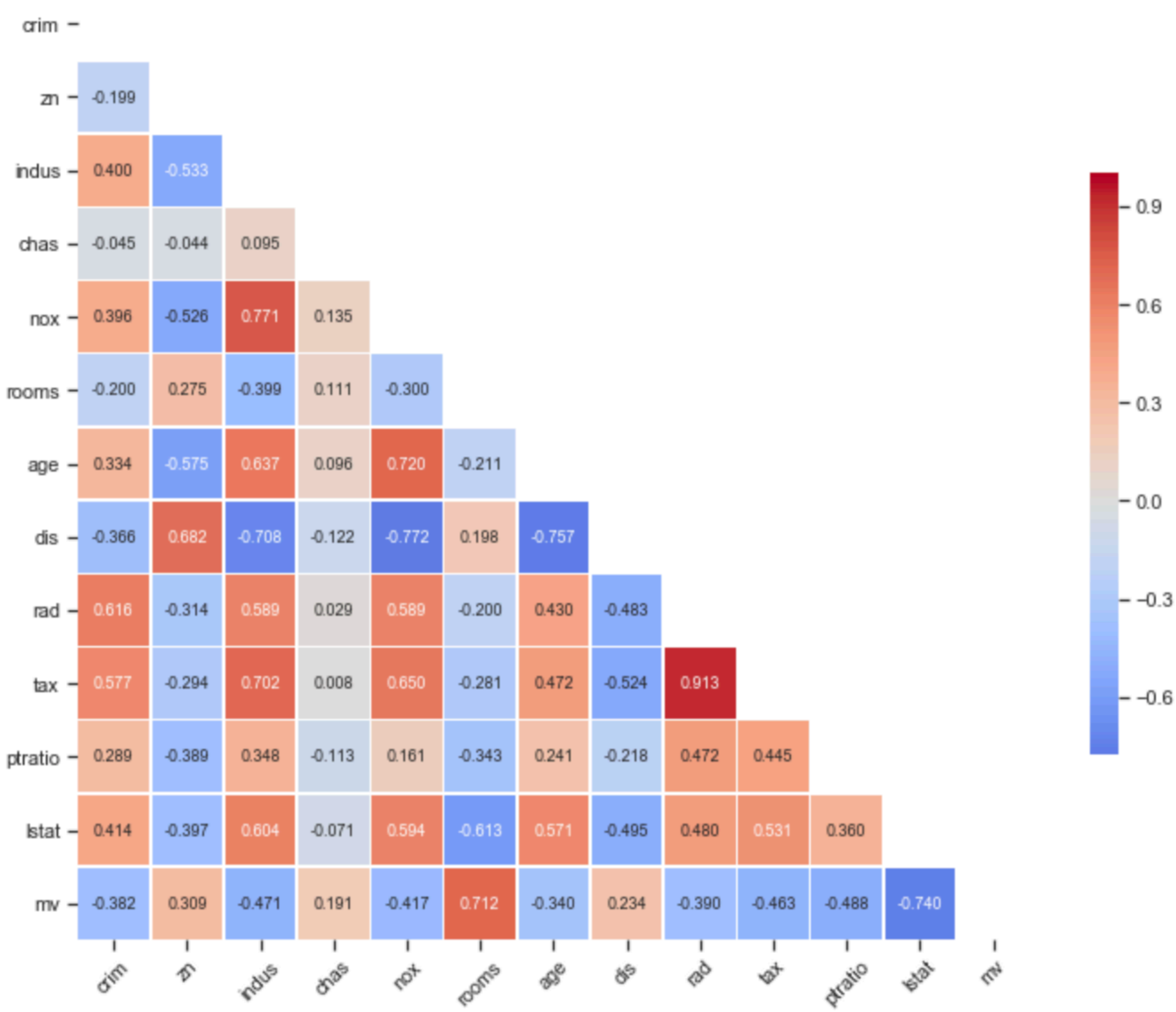


As a final test of our model, *sklearn's staged_predict* method is employed to check on the benfits of stopping before 100 estimators are used. Though **figure 8** shows MSE declining with diminishing returns to additional estimators, the minimum MSE value is not reached before 100 trees are employed.



Management recommendations.

<u>Which model?</u> The Gradient Boosting regressor is certainly not the most transparent model but it is quite powerful and flexible and the results in terms of error reduction are significant versus linear models employed in Assignment 3. Linear models were not the worst of the 80, but their RMSE were much larger than the tree-based models. If management is willing to engage in discussions regarding parameters employed, this model is well suited. Simpler decision trees may be a good stepping stone. Limiting features to just 2, rooms and lstat, may assist the ensuing leap into Random Forests and eventually Gradient Boosting can be accepted as a model for forecasting median values. Without a doubt however tree based models are preferred over linear.


crim
zn
indus
chas
nox
rooms
age
dis
rad
tax
ptratio
lstat
mv
-0.199
0.400 -0.533
-0.045 -0.044 0.095
0.396 -0.526 0.771 0.135
-0.200 0.275 -0.399 0.111 -0.300
0.334 -0.575 0.637 0.096 0.720 -0.211
-0.366 0.682 -0.708 -0.122 -0.772 0.198 -0.757
0.616 -0.314 0.589 0.029 0.589 -0.200 0.430 -0.483
0.577 -0.294 0.702 0.008 0.650 -0.281 0.472 -0.524 0.913
0.289 -0.389 0.348 -0.113 0.161 -0.343 0.241 -0.218 0.472 0.445
0.414 -0.397 0.604 -0.071 0.594 -0.613 0.571 -0.495 0.480 0.531 0.360
-0.382 0.309 -0.471 0.191 -0.417 0.712 -0.340 0.234 -0.390 -0.463 -0.488 -0.740
0.9
0.6
0.3
0.0
−0.3
−0.6